加微信：642945106 发送“ 赠送 ”领取赠送精品课程
发数字“ 2 ”获取众筹列表

03 | SRE切入点：选择SLI，设定SLO

2020-03-23 赵成

SRE实战手册 进入课程 >

讲述：赵成
时长 16:15 大小 14.89M

你好，我是赵成，欢迎回来。

还是先来复习下上节课讲的“系统可用性”的两种计算方式，一种是从故障角度出发，以时长维度对系统进行稳定性评估；另一种是从成功请求占比角度出发，以请求维度对系统进行稳定性评估。同时，我们还讲到，在 SRE 实践中，通常会选择第二种，也就是根据成功请求的比例来衡量稳定性：

Availability = Successful request / Total request

SRE 强调的稳定性，一般不是看单次请求的成功与否，而是看整体情况，所以我们会把成功请求的占比设定为一个可以接受的目标，也就是我们常说的“3 个 9”或“4 个 9”这样

的可量化的数字。

那么，这个“确定成功请求条件，设定达成占比目标”的过程，在 SRE 中就是**设定稳定性衡量标准的 SLI 和 SLO 的过程**。

具体来看下这两个概念。SLI，Service Level Indicator，服务等级指标，其实就是我们选择哪些指标来衡量我们的稳定性。而 SLO，Service Level Objective，服务等级目标，指的就是我们设定的稳定性目标，比如“几个 9”这样的目标。

SLI 和 SLO 这两个概念你一定要牢牢记住，接下来我们会反复讲到它们，因为落地 SRE 的第一步其实就是“**选择合适的 SLI，设定对应的 SLO**”。

好，那我们就正式开始今天的内容。我会带你彻底理解 SLI 和 SLO 这两个概念，并掌握识别 SLI、设定 SLO 的具体方法。

SLI 和 SLO 到底是啥?

SLI 和 SLO 这两个概念比较有意思，看字面意思好像就已经很明白了，但是呢，仔细一想，你会发现它们很抽象。SLI 和 SLO 指的到底是啥呢?

接下来我给你讲一个具体的例子，讲完后，你肯定就能理解了。

我们以电商交易系统中的一个核心应用“购物车”为例，给它取名叫做 trade_cart。trade_cart 是以请求维度来衡量稳定性的，也就是说单次请求如果返回的是非 5xx 的状态码，我们认为该次请求是成功的；如果返回的是 5xx 状态码，如我们常见的 502 或 503，我们就判断这次请求是失败的。

但是，这个状态码只能标识单次请求的场景。我们之前讲过，单次的异常与否并不能代表这个应用是否稳定，所以，我们就要看在一个周期内，所有调用次数的成功率是多少，以此来确定它是否稳定。比如我们给这个“状态码返回为非 5xx 的比例”设定一个目标，如果大于等于 99.95%，我们就认为这个应用是稳定的。

在 SRE 实践中，我们用 SLI 和 SLO 来描述。“状态码为非 5xx 的比例”就是 SLI，“大于等于 99.95%”就是 SLO。说得更直接一点，SLO 是 SLI 要达成的目标。

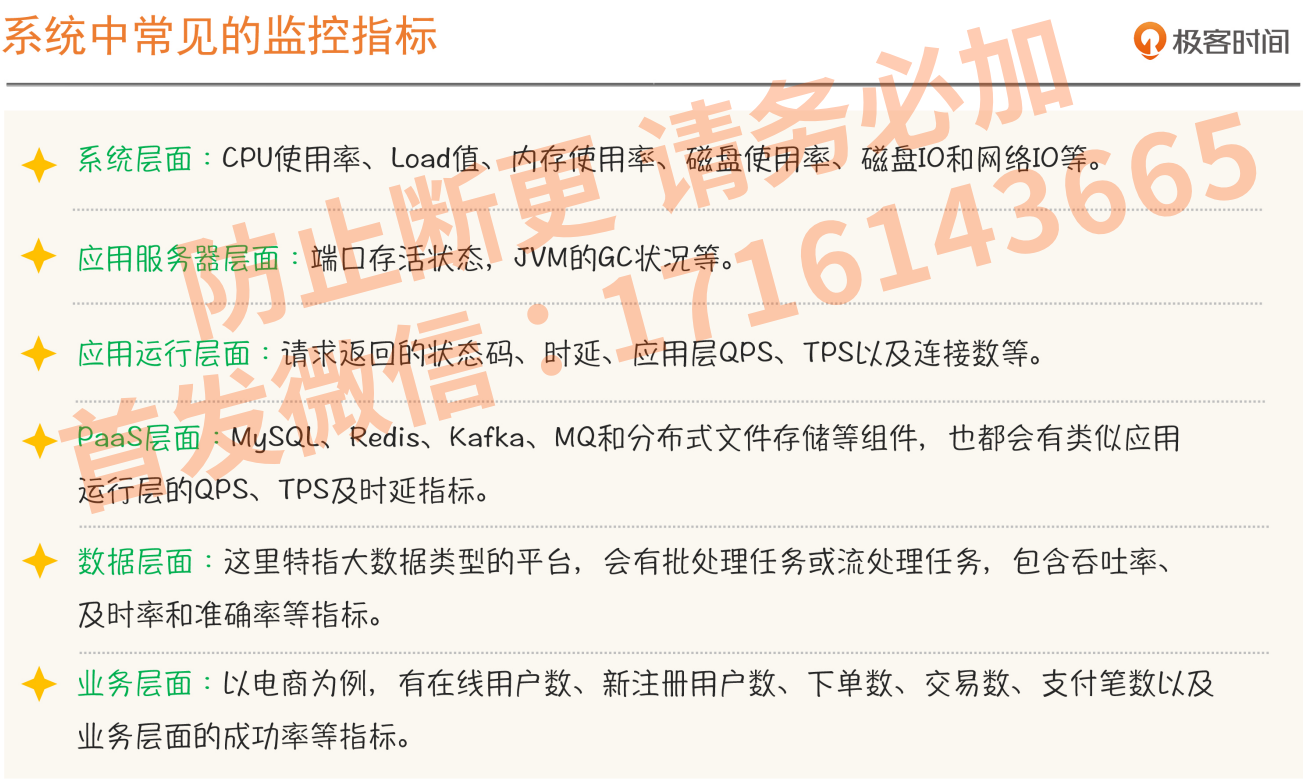
通过这个例子，你现在是不是已经理解了这两个概念呢。SLI 就是我们要监控的指标，SLO 就是这个指标对应的目标。

好，那接下来我们要解决的问题就很具体了。我们应该选择哪些指标来监控系统的稳定性？指标选好后，对应地怎么定它的目标呢？下面咱们就一一来探索。

系统运行状态指标那么多，哪些适合 SLI？

我们先来讨论怎么选择 SLI。要回答怎么选择这个问题，我们得先来看看都有哪些可供选择的指标。

在下面这张图中，我列举了系统中常见的监控指标。



防止断更 请务必加
首发微信：17716143665

这些指标是不是都很熟悉？那该怎么选呢？好像每一个指标都很重要啊！

确实，这些指标都很重要。我们可以通过问自己两个问题来选择。

第一个问题：**我要衡量谁的稳定性？** 也就是先找到稳定性的主体。主体确定后，我们继续问第二问题：**这个指标能够标识这个实例是否稳定吗？** 一般来说，这两个问题解决了，SLI 指标也就确认了。

从我的经验来看，给指标分层非常关键。就像上图那样分层后，再看稳定性主体是属于哪一层的，就可以在这一层里选择适合的指标。但是，你要注意，即便都是应用层的，针对具体的主体，这一层的指标也不是每一个都适合。

根据这几年的实践经验，我总结了选择 SLI 指标的两大原则。

原则一：选择能够标识一个主体是否稳定的指标，如果不是这个主体本身的指标，或者不能标识主体稳定性的，就要排除在外。

原则二：针对电商这类有用户界面的业务系统，优先选择与用户体验强相关或用户可以明显感知的指标。

还拿我们上面 trade_cart 的例子来说，主体确定了，就是 trade_cart，应用层面的，请求返回状态码和时延就是很好的指标，再来检查下它们能否标识的 trade_cart 稳定性，毫无疑问，这两个指标都可以，那么请求返回状态码和时延就可以作为 trade_cart 稳定性的 SLI 指标。

我们换一个指标，CPU 的使用率这个指标适合吗？根据我们刚才说的原则，既然我们关注的是 trade_cart 的运行状况，而 CPU 是系统层的指标，所以，在选择应用层 SLI 的指标时，自然会把 CPU 排除掉。

你可能会说，这样是不是太武断了呀？

我们简单来分析下。假设 CPU 使用率达到了 95%，但是只要 CPU 处理能力足够，状态码成功率可能还是保持在 4 个 9，时延还是在 80ms 以内，用户体验没有受到影响。另外一种情况，假设 CPU 使用率只有 10%，但是可能因为网络超时或中断，导致大量的请求失败，甚至是时延飙升，购物车这个应用的运行状态也不一定是正常的。所以结论就是，CPU 使用率不管是 10% 还是 95%，都不能直接反映 trade_cart 运行是正常还是异常，不适合作为 trade_cart 这样的应用运行稳定性的 SLI 指标。

讲到这里，你可能会问，哎呀，你说的这两个原则我理解了，分层也大概能做到，但是我还是需要做很多详细的分析才能选择出 SLI 指标，有没有什么更便捷、更快速的方法来帮助我选择啊？

嗯，不要着急，还真有这样一套方法。怎么选 SLI，我们可以直接借鉴 Google 的方法：VALET。

快速识别 SLI 指标的方法：VALET

VALET 是 5 个单词的首字母，分别是 Volume、Availability、Latency、Error 和 Ticket。这 5 个单词就是我们选择 SLI 指标的 5 个维度。我们还是结合 trade_cart 这个例子，一起看一下每个维度具体是什么。

Volume- 容量

Volume（容量）是指服务承诺的最大容量是多少。比如，一个应用集群的 QPS、TPS、会话数以及连接数等等，如果我们对日常设定一个目标，就是日常的容量 SLO，对双 11 这样的大促设定一个目标，就是大促 SLO。对于数据平台，我们要看它的吞吐能力，比如每小时能处理的记录数或任务数。

Availablity- 可用性

Availablity（可用性）代表服务是否正常。比如，我们前面介绍到的请求调用的非 5xx 状态码成功率，就可以归于可用性。对于数据平台，我们就看任务的执行成功情况，这个也可以根据不同的任务执行状态码来归类。

拼课微信：1716143665

Latency- 时延

Latency（时延）是说响应是否足够快。这是一个会直接影响用户访问体验的指标。对于任务类的作业，我们会看每个任务是否在规定时间内完成了。

讲到这里，我要延伸下，因为通常对于时延这个指标，我们不会直接做所有请求时延的平均，因为整个时延的分布也符合正态分布，所以通常会以类似“90% 请求的时延 <= 80ms，或者 95% 请求的时延 <=120ms ”这样的方式来设定时延 SLO，熟悉数理统计的同学应该知道，这个 90% 或 95% 我们称之为置信区间。

因为不排除很多请求从业务逻辑层面是不成功的，这时业务逻辑的处理时长就会非常短（可能 10ms），或者出现 404 这样的状态码（可能就 1ms）。从可用性来讲，这些请求也算成功，但是这样的请求会拉低整个均值。

同时，也会出现另一种极端情况，就是某几次请求因为各种原因，导致时延高了，到了500ms，但是因为次数所占比例较低，数据被平均掉了，单纯从平均值来看是没有异常的。但是从实际情况看，有少部分用户的体验其实已经非常糟糕了。所以，为了识别出这种情况，我们就要设定不同的置信区间来找出这样的用户占比，有针对性地解决。

Errors- 错误率

错误率有多少？这里除了 5xx 之外，我们还可以把 4xx 列进来，因为前面我们的服务可用性不错，但是从业务和体验角度，4xx 太多，用户也是不能接受的。

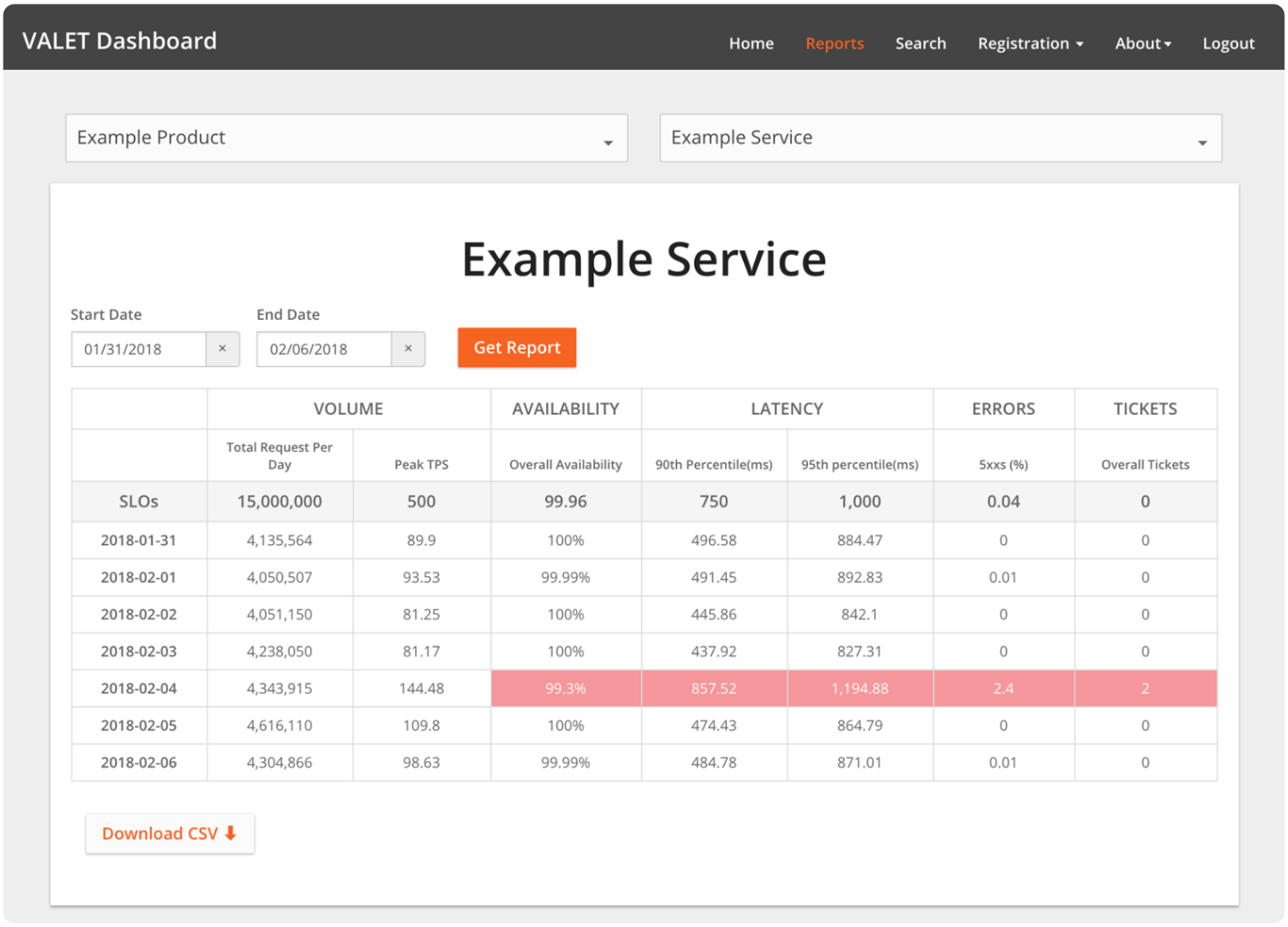
或者可以增加一些自定义的状态码，看哪些状态是对业务有损的，比如某些热门商品总是缺货，用户登录验证码总是输入错误，这些虽不是系统错误，但从业务角度来看，对用户的体验影响还是比较大的。

Tickets- 人工介入

是否需要人工介入？如果一项工作或任务需要人工介入，那说明一定是低效或有问题的。举一个我们常见的场景，数据任务跑失败了，但是无法自动恢复，这时就要人工介入恢复；或者超时了，也需要人工介入，来中断任务、重启拉起来跑等等。

Tickets 的 SLO 可以想象成它的中文含义：门票。一个周期内，门票数量是固定的，比如每月 20 张，每次人工介入，就消耗一张，如果消耗完了，还需要人工介入，那就是不达标了。

这里我给出一个 Google 提供的，针对类似于我们 trade_cart 的一个应用服务，基于 VALET 设计出来的 SLO 的 Dashboard 样例，结合上面我们介绍的部分，就一目了然了。



	VOLUME		AVAILABILITY	LATENCY		ERRORS	TICKETS
	Total Request Per Day	Peak TPS	Overall Availability	90th Percentile(ms)	95th percentile(ms)	5xxs (%)	Overall Tickets
SLOs	15,000,000	500	99.96	750	1,000	0.04	0
2018-01-31	4,135,564	89.9	100%	496.58	884.47	0	0
2018-02-01	4,050,507	93.53	99.99%	491.45	892.83	0.01	0
2018-02-02	4,051,150	81.25	100%	445.86	842.1	0	0
2018-02-03	4,238,050	81.17	100%	437.92	827.31	0	0
2018-02-04	4,343,915	144.48	99.3%	857.52	1,194.88	2.4	2
2018-02-05	4,616,110	109.8	100%	474.43	864.79	0	0
2018-02-06	4,304,866	98.63	99.99%	484.78	871.01	0.01	0

VALET示例图

好，VALET 我们就讲完了，怎么选 SLI 指标，你是不是一下子就清楚了。可以说，这是一个我们可以直接复用的工具。上面 Google 的这张 SLO 样例图，建议你多看几遍，看到时候，对比思考下自己系统的情况。

如何通过 SLO 计算可用性?

到这里，我们已经能够根据自己想要保障稳定性的主体，来选择合适的 SLI 指标了，也知道 SLO 就是对应 SLI 要实现的目标，比如“几个 9”。

但是，我们前面讲到了系统可用性：

Availability = Successful request / Total request

然后又深入到了提炼具体的 SLI，以及设定对应的 SLO，那这两者之间是什么关系呢？也就是通过 SLO 应该怎么去计算我们的系统可用性的呢？这就涉及到系统整体可用性的两种计算方式。

第一种，直接根据成功的定义来计算。

也就是我们前面定义一个请求的返回状态码必须是非 5xx 才算成功，同时时延还要低于 80ms，同时满足这两个条件，我们才算是成功的，也就是纳入上述公式中 Successful request 的统计中，用公式来表示：

Successful = （状态码非 5xx） & （时延 <= 80ms）

如果还有其它条件，直接在后面增加做综合判定。

但是，这种计算方式存在的问题就是，对单次请求的成功与否的判定太过死板，容易错杀误判。比如我们前面讲对于时延，我们一般会设定置信区间，比如 90% 时延小于等于 200ms 这样的场景，用这种方式就很难体现出来。而且，对于状态码成功率和时延成功率的容忍度，通常也是也不一样的，通过这种方式就不够准确。所以，我们就会采取第二种方式。

第二种方式，SLO 方式计算。

我们前面讲时延时讲过以下几个 SLO，这时我们设定稳定性的时候，就需要把公式计算方式灵活调整定义一下了。

SLO1：99.95% 状态码成功率

SLO2：90% Latency <= 80ms

SLO3：99% Latency <= 200ms

直接用公式表示：

Availability = SLO1 & SLO2 & SLO3

只有当这个三个 SLO 同时达标时，整个系统的稳定性才算达标，有一个不达标就不算达标，这样就可以很好地将 SLO 设定的合理性与最终可用性结合了起来。所以，通常在 SRE 实践中，我们通常会采用这种设定方式。

如果是这样，第一种方式是不是就没有用途了呢？当然不是。第一种计算方式也会有它特有的应用场景，它通常会被利用在第三方提供的服务承诺中，也就是 SLA（Service Level Agreement）。因为对于第三方提供商来说，比如云厂商，它们要面对的客户群体是非常大的，所以很难跟每一家客户都去制定像 SLO 这么细粒度稳定性目标，而且每家客户对稳定性的要求和感知也不同，就没法统一。

这种情况下，反而是第一种计算方式是相对简单直接的，但是这样也决定了 SLA 的承诺相比 SLO 肯定也相对比较宽松，因为 SLA 是商业服务承诺，如果达不成是要进行赔偿的。关于 SLA，最直接的参考资料，就是各个公有云厂商在官网公开的信息资料，你可以找到这些资料，作为自己的一个补充学习。

总结

讲到这里，怎么选择 SLI 指标、如何制定 SLO 目标，我们就介绍完了。你需要掌握下面三个重点。

1. 对系统相关指标要分层，识别出我们要保障稳定性的主体（系统、业务或应用）是什么，然后基于这个主体来选择合适的 SLI 指标。
2. 不是所有的指标都是适合做 SLI 指标，它一定要能够直接体现和反映主体的稳定性状态。可以优先选择用户或使用者能感受到的体验类指标，比如时延、可用性、错误率等。
3. 掌握 VALET 方法，快速选择 SLI 指标。

思考题

最后，给你留一个思考题。

下面我给出一个 Google 的 SLI 和 SLO 设定标准示例，内容很直观，需要你认真研究一下这个文档，结合今天我们所讲的内容，请你尝试按照 Google 提供的规范格式，制定一个自己所负责系统的 SLO。

Google 的 SLI 和 SLO 设定模板链接：
https://landing.google.com/sre/workbook/chapters/slo-document

另外，对今天的内容如果你还有什么疑惑，都可以在留言区提问，也欢迎你把今天的内容分享给身边的朋友，和他一起学习讨论。

我是赵成，我们下节课见。

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇　02 | 系统可用性：没有故障，系统就一定是稳定的吗？

精选留言 (2)

写留言

leslie
2020-03-23

老师分享的标准要翻墙、、、我就基于课程内容谈谈吧，觉得课程中的东西结合之前运维体系课程可以理解缘由。
“SLO 是 SLI 要达成的目标。”一个是等级一个是分数；就如同“Availability = SLO1 & SLO2 & SLO3”公式一样；这就如同行业等级测评一样，到达某个等级必须多项条件都只要达到一个分数。...

展开

作者回复: 分享的很好。系统打分是基于系统运行过程中的多个指标综合评判得出的。

云厂商定的标准我们一般称之为SLA，不一定是错误，但肯定是相对简单和通用一些，因为它是要面向更大范围的客户的，定的太过精细反而不易于执行。

1

Nick

2020-03-23

通过SLI和SLO, 是不是只能判断在一个时间范围内是否都达标. 但没法做到像前面的那个时间纬度那样的3条9和4条9这样的表. SaaS的服务承诺好像是用类似时间维度那样的3条9, 4条9这样的故障时间评判表.